Toples

LA 99ers COMPUTER GROUP Newsletter

VOL. 5 NO. 11 *LOS ANGELES, CA* NOV. 1984

TERRIES'S CORNER

Teresa Masters President

POTPURRI

A tale of two cities. I attended both the Chicago and Milwaukee "FAIRES". Last year several of us returned home feeling we had missed something very important. Primarily the failure to locate and meet persons we had come to know by deed alone. This year we tried to rectify that situation on our own. I personally went over to names on tags saying "I want to meet you". In this way I found Clint Pulley, a giant in "C" language. I sorely missed John Taylor, John Clulow and Chris Faherty. They did not attend this year. Some of us had communicated in advance and were able to find one another this way. Rory Binkerd and Bob Daggett, Walt Howe, Peter Hoddie and Cynthia Becker. Ron Albright made his first physical appearance. Names well known to Source, Compuserve and Genie were very much in evidence. Blaine Crandall, Barry Traver, Dee Turner. A couple of folks who take exception to some of my positions, or the way I express them, Coe Case, and Art Byers. My good friend and occasional defender, Howie Rosenberg. Publisher Richard Mitchell. (John Koloen was there again and again I missed him). Bob Boone and his Ottawa group. Jim Horn, Jeff Guide of CIS and DOS. Warren Agee and a very kind Todd Kaplan, who enabled Warren to spend more time among the gathering by volunteering a 500 mile drive. John Behnke local programmer of merit. Asgard's Chris Bobbitt. Lou Phillips with Paul Charlton and GENEVE AKA 9640. A couple of others I just do not feel like turning the other cheek for. Some I may have unintentionally just missed, sorry. Then there was JIM PETERSON, TIGERCUB. What a delightful man! Undoubtedly the MOST undersung supporter of the 99/4A community. I have felt the urge over the last few years to simply telephone Jim and let him know we APPRECIATE him. The imagination and creativity Jim has put into the XBasic language is unmatched. Too many persons as they "grew" beyond XB have then looked down at this excellent language. Jim has given it a hell of a go. His Tigercub has been the mainstay of many a Newsletter. I wonder just how often he has been really thanked. Peter Hoddie was extremely complimentary of Jim's efforts during his extraordinary demonstration. He made quite a few people aware of Jim's value. Time for more of us to let Jim know how well respected and appreciated he is.

That essentially was Chicago, except for a very LARGE thank you to Dean Conrad our own LA 99er now transplanted to Chicago. Dean and Pam Conrad were Hosts extraodinaire to a bunch of us. Barry Traver and Son John Calvin, Howie Rosenberg, Ron Albright, Richard Mitchell and me. Dean was also our Tour Guide and Escort to Milwaukee. What a great couple. Thank you very much.

Milwaukee, what a nice warm bunch of people. The Faire was smaller, much more intimate, and we all felt very welcome. The prices were for the most far more reasonable. I spent more in the first 30 minutes there than in the whole day in Chicago. One very sour note and a large BLACK HAT to SYMBIOTECH. Discourtesy and uncaring arrogance was observed by many of us unfortunately adjacent to his table. I personally was purchasing some MBX modules, not yet priced. When asking prices, was told in a very annoyed manner to look in the catalog in the box marked "free take one". When asked why one cataloged at $7.50 was charged at $9.00, I was told that was catalog price and the other was because "it cost me money to get here". In also asking why the great variance in price on these modules was told "as availability goes down, prices go up". Nice guy.

I was told over that weekend that we had become component mentality purchasers. Looking around the booths I noticed the interest in both the Rave Keyboard and the 80 column card. The cost of the two items combined appear to be within $25.00 of the Geneve 9640 system, which includes both with many other extras, 640K for example.

TI, in its continuing support efforts, 3 years after Black Friday, has released to User Groups a complete set of diagnostic support material. The efforts of St. Louis User Group President Knapp were very instrumental in this. Thanks to a job well done. Willing, responsible persons within User Groups can now attempt to diagnose problems, and perhaps resolve them locally. This will mean only the unsolvable ones get shipped to Lubbock. This package is clearly marked "not for sale", hopefully they will not appear in catalogs with a PRICE tag.

The L.A. 99er Group prize from the Seattle "FEST" was a years subscription to Micropendium. This will be included in a Club raffle shortly. Someone remind me please.

Our other friendly Doctor is at it again. Ron Albright is once again publishing. "ORPHAN'S SURVIVAL HANDBOOK" is in its final stages and should be available shortly. I have reviewed the first 130 pages, yes it will be a BIG one. The material contained within will be of value to everyone. It is fruition of a concept discussed for a couple of years. I totally support this effort by Ron. L.A. 99er Marketplace will include this publication and hopefully we will be able to discount it. More later.

UGN revitalized. As an adjunct to the above, future additions will be made available, perhaps quarterly to all owners of "ORPHAN'S SURVIVAL HANDBOOK". I will need lots of help in this project and will be contacting User Groups shortly. It was suggested to me that UGN might not be the best name for this endeavor, as there are people who are active in the community and do not belong to a User Group. My brain has not yet developed a better name. Any suggestions?

My compliments to Lou Phillips for just honestly being there and calmly answering all questions, some VERY loaded. I have lately seen a definite turning of the tide. There are still some rough edges, but slowly as each wave ebbs they are becoming smooth. For someone who has had dreams battered, it feels good to see this dream looking so very good. What an incredible group of young talent now stand with Myarc. Vaya con Dios.

New Programs, the last couple of months have blessed us with some new and exceptionally fine programs. Three by Peter Hoddie, PRE-SCAN IT!, GRAM PACKER, and FONT WRITER. Mike McCann's PRINTER'S APPRENTICE, and Danny Michael's GRAM UTILITY for MG. Not bad for an ORPHAN. Look in MARKETPLACE for all. We have them in hand. Continued support of these fine efforts will result in their continued support of us.

To those of you who took the time to respond, and sent George Steffen words of Hope, Thank you very much. I saw the light in George's eyes and heart when he received them.

L.A. members, a nominating committee has been appointed, elections are coming up in a few months, time to evaluate what you can do for your club. We are a strong club, with an excellent Library run by an exceptional librarian, a highly ranked Newsletter thanks to the efforts of several, good meetings thanks to an infusion of new efforts, financially ok because of ongoing pursuits to keep us that way. All things work hand in hand in cooperation, this is a nice foundation upon which to build. Go for it!

**********NOVEMBER MEETING**********

The meeting will begin with the President's report by Terrie Masters. The Library will be represented by Chick DeMarti discussing the latest additions. Following Chick's report, Gail Fair will discuss Marketplace Specials. Well we finally have the answer to the question you have been asking. How do I manage all my diskettes and documentation? George Hutton has found the answer and he will present a short demonstration of Origami. That is the art of programer paper folding. George's technique will assist you in keeping your software organized. After the social break, there will be a question and answer session. The sessions we have held at previous meetings were very popular. Many interesting questions were asked, but unfortunately the time allocated was too short for all the questions to be answered. More time will be devoted at this meeting in order to catch up on the unanswered questions and ask new ones. Bring your questions, comments or ideas about the TI99/4A to share with the club. The November meeting will be hosted by Steve Chalcraft.

**********CLUB OFFICERS**********

President		Membership Chairman	
Terrie Masters	(213) 271-6930	Tom Freeman	(213) 454-1945
Vice President		Librarian	
George F. Steffen	(213) 329-3527	Fred Moore	(213) 670-4293
Secretary		Library Assistant	
Terry Wilson	(213) 563-3869	Chick DeMarti	(213) 532-8499
Treasurer		Library Assistant	
Margaret Hutton	(213) 541-5359	Allen Whiteman	(213) 370-8031
Editor		Equipment Chairman	
Terrie Masters	(213) 271-6930	Joe Fierstein	(213) 377-9834

Topics solicits advertisers at the following rates:
FULL PAGE $20.00
HALF PAGE $12.00
QUARTER " $ 8.00

MEETINGS
NOVEMBER 26
DECEMBER 18
JANUARY 28
FEST-WEST'87 MAY 2 & 3

A PERSONAL NOTE by George F. Steffen

I wish to express my thanks to the many people who sent prayers, flowers, cards and just well wishes during my recent hospitalization. I am still undergoinG radiation therapy and probably will have an operation in the near future. I am feeling well and expect a complete recovery. Unfortunately, the effects of the radiation make it difficult for me to sit for extended periods and thus have cut down on my programming time.

In view of my physical condition and the possibility of a move to the East, I request that I not be nominated for any officer position in the club. I intend to maintain my membership and to continue programming and writing for the newsletter as much as I can.

CORCOMP DISK CONTROLLER MODIFICATION
by George F. Steffen

Millers Graphics has released a set of modified EPROMs for the CORCOMP Disk Controller. These will allow users of this controller to bypass the title screen and go automatically to one of the main menu choices (see Gram Kracker manual) as could users of other disk controllers. It adds CALLs to LOAD and RUN assembly language programs and allows these CALLs and the Toolshed Utilities to be accessed from a running XB program. The revision also includes some other enhancements to the controller.

Since I have a battery backed RAMdisk which is configured as Disk 5, I have changed my Extended Basic in the Gram Kracker to automatically run a program called GFSL on Disk 5. This program is a short program which I wrote myself which offers all options (Basic, Extended Basic, Hard Disk Controller, Floppy Controller or Funnelwriter (which will ' giPe me access to TI Writer and Editor/Assembler). Thus, when I turn on my system, I am immediately presented with my favorite options without having to hit the 2 key twice to get XB and then wait for the program to load. The EPROM set costs $34.95 and is available through our group.

GK UTILITY I, a review by Mike Dodd, K/Town 99ers.
(FROM K/TOWN 99ER)

Performance.....A+
Ease of Use.....A+
Documentation.....A
Value.....A+ (or higher)
Final Grade.....A+
Cost: $10, including shipping
Manufacturer: Millers Graphics, 1475 W. Cypress Ave, San Dimas, CA 91773

Requirements: TI99/4A (not 99/4), 32K memory expansion, disk system, version 110 of TI Extended Basic, TI Editor/Assembler, and an 80K Gram Kracker.

When tHe Gram Kracker was first announced in September 1985, programmers everywhere were ecstatic. Now they could modify the cartridges that TI put out. Since the first GKs were received, in late December, programmers have been changing the mysterious TI cartridges. Many of these 'mods were published in newsletters. Some were uploaded to CompuServe, The Source, or GEnie. Now, Danny Michael has come up with a complete set of modifications to Extended Basic and to the Editor/Assembler, all in one package. Now dozens of enhancements to XB and E/A can be made in a matter of seconds, all for a mere $10.

PERFORMANCE: The disk makes many modifications to the Extended Basic module, including:
CALL INIT... XB loaded more data into memory expansion than was needed. This has been fixed with this disk.
LIST..... If you are listing to disk or printer, you may now specify a line lngth. The old XB LISTed in 80 column format. Now you can list in 28 columns, 132 columns, anything. To list lines 100-250 to PIO in 130 columns, you would type LIST "PIO":130:100-250.
CALL LOAD... If you use this to "poke" data into RAM (i.e. CALL LOAD(8192,0)), it no longer checks to see if CALL INIT has been performed. If you load an assembly object file, it still makes sure CALL INIT has been executed.
RESequence.. You may now resequence a portion of the program. To resequence lines 1000-2000 to start at line 800 with an increment of 5, you would use RES 800,5,1000-2000. You cannot use RES to move lines from one section of the program to another. If you attempt to do so, it will result in an error.
TRACE..... You can not dump a TRACEing out to a printer. To do so, include a line in your program consisting of OPEN #123:"printer name". While this is more inconvenient than TRACE "PIO" or some such, it does have sme advantages. You can, for instance, have your program print out where it is in the program. Example:

```
1 OPEN #123:"PIO" :: TRACE :: PRINT #123:"Main program"
100 ! MAIN
110 ! PROGRAM
120 ! GOES
130 ! HERE
140 CALL SAVE(A$)
    .
    .
800 SUB SAVE(A$):: PRINT #123: : "Saving to disk"
```

900 PRINT #123: "Done Saving" :: SUBEND

New commands:

COPY..... You can copy a section of a program from one area to another. For instance, to copy lines 1000-1400 to line 3000 with an increment of 4, you would typ COPY 1000-1400,3000,4.
MOVE..... Same as above, only it deletes the source lines after it copies them.
DEL..... Deletes a block of lines. DEL 1340-7890 will delete all lines from 1340 through 7090, inclusive.
CALL EA..... Pass control to the Editor/Assembler module, assuming you have chosen the combined XB and EA when you ran the patch program.
CALL PEEKG/POKEG..... These subroutines peek and poke data from and to Grom/Gram. Note that you can only poke to Gram 1 or 2, as all other Grams are protected in XBasic.
CALL PEEKV/POKEV..... These subroutines peek and poke data to VDP RAM.
CALL QUITON/QUITOFF.. These routines enable and disable the quit key.

Other modifications:

The screen color has been changed to white on dark blue.
The cursor character has been redefined to an underline.
All error messages now appear in upper/lower case.
On entry to XBasic, the quit key will be disabled
If you hold down a key on entry to XB, it will no search for the LOAD file on disk.
Additional cursor movement keys have been added. In XBasic, often you will have a five-line program line. If you want to get to a point a few lines down, you have to hold down FCTN D for a LONG time. Now, FCTN-SHIFT X and FCTN-SHIFT E will move the cursor up ad down on the line. FCTN-SHIFT D and FCTN-SHIFT S will move the cursor to the end of the line and the beginning of the line respectively. These keys will also work when responding to an INPUT or ACCEPT statement in a running XB program.

EDITOR/ASSEMBLER MODS:

All titles and prompts are now in Upper/lower case.
Auto-repeat has been added. FCTN 3 (Erase line) is now active. FCTN 4 (clear) will now clear from the cursor t the end of the line. FCTN-SHIFT S and FCTN-SHIFT D will place the cursor at the beginning and end of the line, respectively.
All filenames are now stored in Gram 3 for recall. Filenames entered in LOAD RUN, RUN PROGRAM FILE, the Editor, and the Assembler are remembered, even when you turn the computer off.
The Editor and the Assembler are now stored in Gram, instead of on the disk, for instant loading. The assembler is stored in Gram 1-2, and the editor is scattered throughout XBasic.
New menu selections:
6 - Extended Basic: Goes to XBasic instantly.

7 - Format RAMdisk: Will execute a CALL PART(096,000) and CALL EMDK(5) command, if you have a MYARC 128/512 card. Instructions are given for changing the values.

8 - Catalog Disk: Catalogs a disk to the screen.

There is one disadvantage to theis E/A - the basic support CALLs (CALL INIT, LOAD, LINK, PEEKV, POKEV, CHARPAT) have been deleted. This was required for the modifications Danny made, but it would have been nice if they could have been preserved somehow.

The letter "O" is rounded and the number "0" is slashed. It installs a CREAT looking character set.

EASE OF USE: To make the modifications, load XBEAPATCH with Load Module. It will ask you if you want XBasic only, or combined XB and E/A. If you select XB and E/A, it will ask you to place the disk with the EDIT1, ASSM1, ASSM2 files in DSK1. It then starts patching the modules. This takes about 8 seconds. This is incredibly easy. Using the utilities is also very easy, as Danny kept the standard TI format in all his new commands. You shold Have no trouble remembering it.

DOCUMENTATION: The documentation is excellent. consisting of 22 pages plus a two page table of contents. It includes complete instructions and explanations for all the mods, and a five page technical section that includes addresses of important bytes, and a list of all free areas in memory. The documentation comes on a disk, which is fine except that, if you don't have a printer, you're out of luck. That is the only reason the docs got an "A" inwtead of an "A+". Although, if he included printed docs, the price would have been much higher.

VALUE: My only regret is that I couldn't give this a rating highter than "A+". Ten dollars is an incredible price for a disk of this qualiy.

OVERALL: If you own a Gram Kracker, there is no reason in the world not to buy this disk. Even if you don't, you might want to get the GK just for this disk,. It's like a whole new XBasic.

ALL ABOUT PROGRAM FILES

[Ed. Note: The following article appeared in the Ozark 99 User Group Newsletter and was sent to us by Steve Langguth. It originally contained a third part on using the SAVE utility, but as I had written one along these lines some time ago, I decided to revise and expand my own, and include it instead. Tom Freeman]

This article was compiled from three other articles appearing in other newsletters. The first article was on "Program Files" and was written by R.A. Green of the Ottawa 99/4 Users Group. It originally appeared in the May 1986 newsletter of the New Jersey Users Group and was reprinted in the September 1986 newsletter of the Kansas City TI 99/4A Users Group. The second article explained the "TI Program Image Format" in more detail and was written by Jerome Trinkl. It first appeared in the April 1986 newsletter of the Atlanta 99/4A Computer Users Group and was reprinted in the September 1986 newsletter of the Greater Akron (Ohio) 99ers.

There are seven different ways to store programs in the TI 99/4A. In this article we will first look at each of these seven forms and how they are used and then focus in on one of the most useful formats.

Most everyone is familiar with the form used by TI Basic to store programs on cassette or disk. It's identified as "PROGRAM" when a disk is cataloged. It is created or stored by the Basic "SAVE" command and loaded into the computer by the Basic "OLD" command. This is the only way that TI Basic uses to store programs.

Extended Basic can, and usually does, use the same format as console Basic to store programs. There are, however, two other forms that Extended Basic can use to store programs on disk (but not cassette). If you have the 32K Memory Expansion, you can write an XB program which is too large to store in the usual format. XB will store these large programs in an "INTERNAL VARIABLE 254" file. The usual "SAVE" and "OLD" commands are used to store and load these programs. The third form used by XB is the "merge format" stored in a "DISPLAY VARIABLE 163" file. This form is created when the "MERGE" option is specified with the "SAVE" command. The beauty of merge format is that when it is loaded it does not necessarily overwrite the program in memory. The "MERGE" command does just that—it merges the new program (or program segment) with the program in memory according to the line numbers. (A line from a file being "MERGE"d that has the same number as one already in memory will overwrite the old line.)

Now, we get to the "good stuff", Assembly Language programs. There are three formats for A/L programs : tagged object, compressed tagged object, and memory image.

Tagged object code files are stored in "DISPLAY FIXED 80" files on disk only. All program data is in hexadecimal so that it can be edited by the E/A editor. Tagged object code can be loaded by "CALL LOAD" in XB, option 3 ("Load and Run") on the E/A menu, option 1 on the MiniMemory module menu (boy, that's a real tongue twister!!!), or by "CALL LOAD" in TI Basic when either the E/A or Minimem module is plugged in. The programs can have "absolute addresses" or be "relocatable". A program with an "absolute address" is always loaded into the same place in memory. A relocatable program can be loaded into any place in memory. A tagged object program can have references to other programs or subroutines. The loader will resolve these "external references", except for the XB loader.

Compressed tagged object code is very nearly the same as tagged object code except that the program data is stored as bytes rather than as hexidecimal digits. Compressed tagged object code loads faster than regular tagged object code, as you would expect. The XB loader cannot load compressed object code.

Tagged object code, in either regular or compressed form (compressed if the "C" option is chosen while assembling) can be produced by the Assembler when it "assembles" source code.

The "memory image" format of assembly language program is the most compact and the fastest loading. It can be stored on cassette or disk. It is identified as "PROGRAM" in a disk catalog (just like a Basic program). Memory image programs can be loaded by option 5 on the E/A module menu or option 3 on the TI Writer module menu. It should be noted that there is one small difference between how the E/A calls a memory image program and how TI Writer does it. TI Writer blanks the screen just before calling the program and E/A does not. This means the program being loaded must turn the screen back on or nothing will show. Memory image programs are created by a utility program, like the one called "Save" that is provided on the E/A disk.

A "PROGRAM" file, whether it contains an Assembler memory image or a Basic program, can be read from or written to any input/output device with a single I/O operation. This is one of the reasons that they load so quickly.

There is a limit on the size of an Assembler memory image file of >2000 bytes. (The reason for this limit is that the various loaders use the 16K VDP memory to transfer the data during the Device Service Routine, and only that number of bytes have been allocated in the transfer buffer.) However, the E/A and Ti Writer modules will load multiple memory image files to make up a program of any length. They use the convention that the file name of the second and following files is obtained by incrementing the last digit or letter of the previous file name. For example, the TI Writer editor consists of two memory image files, "EDITA1" and "EDITA2".

(As a matter of interest...the Adventure, Tunnels of Doom, Personal Record Keeping, Statistics, and Personal Report Generator modules all use memory image files to store data. The fact that memory image files can be saved or loaded with a single I/O operation makes them attractive for such uses.)

to memory image format. See NOTE 2 below.

Note down the first and last addresses. If the file is RORG, add >A000 to each. Note whether there are REFs to the VDP utilities, in which case see NOTE 1 below. And lastly note whether there is an auto start or not. In this case, if the file is compressed, go on to D) below. If it is uncompressed then load it into the TIW or EA editor. Scan down to the end where a line begins with a 1xxxx or 2xxxx and delete this line. Then resave the file (in TIW, use PF, then F DSKx.filename - in EA, 3 SAVE, N for DIS/VAR 80? prompt). Now go to 3)

B) Without DISkASSEMBLER

Load the DIS/FIX 80 file into the TIW or EA editor. You may get an error message "control characters removed" in EA but don't worry about that just yet. Just press enter then 2 EDIT. If you see lots of blank spaces in EA or control characters in TIW then the file is compressed. Some of the work you do will have to be done with a sector editor such as Advanced Diagnostics or DISK+AID but while you're here scan down to the end of the file where you see a : at the beginning of a line. This is the end of the file, and is preceded by any external REFs and DEFs with readable blocks of 6 characters (spacing always pads the name to 6). If you see any names of VDP utilities you will have to prepare a special file covered in NOTE 1 below. Note whether SFIRST, SLAST, and SLOAD have been defined here. Examine the line above these or the one above the : if there were none. If it begins with a 1 or a 2 then this is an auto start file and will have to be modified. If the file is uncompressed delete the line then resave it (in TIW, use PF, then F DSKx.filename , in EA, 3 SAVE, N for DIS/VAR 80? prompt). If the file is compressed go on to C).

Now go back to the file if it is uncompressed and return to the first line. You will see a 0 followed by 4 characters which are the ASCII representation of the number of bytes of relocatable code in Hex. This is followed by an identifier of 8 characters (it may be spaces, or padded to 8 with spaces). In columns 14-18 you will see either 9xxxx or A0000. In the first case the code is absolute origin at address xxxx. In the second it is relocatable and will load at A000. Note down the value in either case. If the code is relocatable you may add the hex number that followed the 0 at the start of the line to >A000 to obtain the last address used. For absolute code scan down the beginning of each line. Each should begin with a 9yyyy where yyyy is the start address of that line. When you get to the last line of code you have almost the last address. Just add to it 2 for each group of 5 characters after the 9yyyy until you get to a 7F near the end of the line. You now have the first and last addresses which will be used below.

C) Compressed file -Address Range- No DISkASSEMBLER

On your backup disk find the first sector (it should be 34, or 32 in older MYARC FDC's). Remember that each "line" or record begins on byte 0, 80, or 160 of a sector. The first line should start (in Hex not ASCII) 01xxxx and then 16 characters which are the identifier (they are readable in ASCII, as 8 characters). xxxx represents the number of bytes of relocatable code. Note it down. After the identifier you will see 39yyyy if the code is AORG or 41yyyy if it is RORG. Write yyyy down if the code is AORG, or A000 if it is RORG. If the file is RORG you may add the xxxx found at the start of the line to A000 to get the last address. Otherwise scan down sector by sector until you get to the last line before the REFs and DEFs or auto start, in other words the last line beginning with a 39 in Hex. The next line will start either with a hex number from 31 to 36 or with a 3A (Hex for :). This is the first address of the last line of code. Now add 2 to the yyyy after the 39 for each group of 6 characters until you get to the 46 at the end of the "line." You now have the last AORG address. Write down the first and last addresses whether AORG or RORG - they will be used below. Note there are a few strange files that apparently were assemabled with a different assembler from the one TI supplied us, and each line does not begin with an address. In these it will be almost impossible to determine the last address without DISkASSEMBLER.

D) Compressed file -auto start- No DISkASSEMBLER

With your sector editor go to the last sector or two and find the line that begins (in Hex) 31xxxx or 32xxxx. Change the 31 or 32 to 46 and write it to the disk.

E) Other auto starts

A few sneaky programmers auto start their programs not with the above method but instead by inserting the start address into the user interrupt hook at >83C4. If you have a file that auto starts but can't find the 1 or 2 (in ASCII, 31 or 32 in Hex) this may be the method. Look at the end of the last line of code for the following: (compressed) 39 83 C4 42 xx xx, (uncompressed) 983C4Bxxxx. If you see it xxxx is the start address. Replace the 39 with a 46 if the file is compressed, or the 983C4 with 8F000 if it is uncompressed, and th program won't auto start anymore. It may not start at all, but that doesn't matter because we don't want it to! We just want to load it, then convert it.

3) The First Executable Instruction

For the EA #5 loader to work (and all loaders based on it) the initial code must be an executable instruction. If you have a file already in memory image format you can examine the code after the first 6 bytes and see what I mean. It frequently is a B @>xxxx (0460 xxxx is the actual code) or JMP xxxx (10xx) where xxxx is the actual start of the program. Or it may be a normal sequence of code e.g. MOV 11,@>xxxx LWPI yyyy etc. indicating the programmer anticipated saving in this format. If your file does not begin this way there will have to be some additional preparation.

If your file is auto start and you have determined it is done by one of the two methods above then you know what the start address is. If the file is not auto start you should know from the instructions for the program what the name of the start address is (for RUN after LOAD, or CALL LINK in EA Basic) and you can look for it at the end of the file (with a sector editor if the file is compressed). You would see something like 5xxxxSTART (uncompressed) or 35xxxx5354415154 (compressed, reading in Hex). These are relocatable start addresses and the xxxx should be added to A000. Absolute address have a 6

or 36 before the xxxx. If you used DISkASSEMBLER the start address was displayed for you. NOW see whether the start address is also the first address of the file. If it is you are in luck and may proceed on to the next step. If not you may still proceed, but when you are finished see NOTE 3 because further modifications are to be made.

4) SFIRST SLAST file

If your file already has SFIRST, SLAST, and SLOAD in DEFs in it, the programer anticipated using this method and you may go on to step 5). If only one or two of these names are used check to make sure that they are the first, last, and first addresses respectively. If they are, then eliminate the appropriate ones from the file below. If not use a sector editor to change any letter in the name (and type 8 over the 7 at the end if the file is uncompressed).

Now prepare the following special file, using the EA editor. Lines 2-4 should abut the left margin.

```
      DEF SFIRST,SLAST,SLOAD
SFIRST EQU  >xxxx
SLOAD  EQU  >xxxx
SLAST  EQU  >yyyy
      END
```

Here xxxx and yyyy are the first and last addresses determined above. Save the file in DIS/VAR 80 format, then go to the assembler and assemble it using the file just saved as source file and a different name for object code. For List File and Options just press enter. You should rapidly get the 0000 errors message.

5) The Reassembly

Now that you are all prepared the final job is easy. Using the 3 LOAD & RUN option of EA load your file (modified if necessary to eliminate the auto start), the file prepared in 4) if it was needed, and SAVE from the EA utility disk. When the cursor appears again, press enter, type in SAVE for program name, then follow the screen prompts. For purposes of neatness choose a name for the output file that ends in a 1, since 33 sector blocks will be created and each successive one will have the last character increased by 1.

The newly created file should run in EA #5. It won't if there were REFs to the VDP utilities, or if the actual addresses were inserted in the original source code. In this case, see NOTE 1 at the end of the article.

Please note that the file you are converting should either be all RORG in which case it will load at >A000, or it should AORG at >A000 or higher. If it AORGs in the>2000 to >4000 range and above >A000 you should save the two parts separately (create two files in 4) above) using a file name ending in 2 for part above >A000. Furthermore the range >2000 to >2FB0 cannot be used since this is where SAVE loads, unless that area is really only a BSS type block. In DISkASSEMBLER this would be indicated by a series of AORGs without DATA. If you are examining the actual DIS/FIX 80 file in the editor or with a sector editor you would have to see 39xxxx (compressed, in Hex) or 9xxxx (uncompressed, in ASCII) carrying you past >2FB0 with no 42's or B's in between for this to be true. In this case SAVE itself would be saved, and overwritten when the program runs, but that is OK because it isn't needed anymore. Furthermore the program can't use the range between >2000 and >2676 because the EA loader and utilities reside here. If the program appears to do that it was meant to be loaded by some other loader, such as Mini Memory, so something else will have to be done.

All these problems can be fixed up if you have the SAVE source code since it can be AORG'd wherever you want it, and therefore not interfere with the original program. If you have DISkASSEMBLER this can be done by following the instructions in the appendix (naturally I would love it if you would buy a copy!) For those that don't have it I am placing the source code in the club library. The disk can be purchased for $5.00 - all proceeds to the Club, not me! You then place an AORG in it that gives you >800 bytes outside the range of the program to be converted and reassemble it. If the file to be converted is to be in the >2000 range then you must use the Mini Memory cartridge to load it, and an RORG assembled SAVE can be used.

NOTE 1 The EA Utilities

Normally EA #5 programs should stand alone since the utilities are not loaded in first, as they are with #3 type files. There is a way around this however. If the file had REFs to the VDP utilities you know this will be necessary. If there weren't such REFs but the converted program won't run then perhaps there were uses of the actual addresses in the program and you can try this method.

Prepare a short file as in step 4) using >2000 as xxxx and >2676 as yyyy. Assemble it then proceed to step 5) and use a filename such as UTIL0 as the output file. Find the file on disk with your sector editor, and change the first two bytes from 0000 to FFFF. This is a generic file and may be used with all converted programs that need it - all you have to do is copy it to the disk containing the converted program and change the name to one the same as the others but with the last character decreased by 1. I prepared DISK+AID in this manner; the converted files are called DISKAID0, DISKAID1, and DISKAID2.

If your file contained code between >2676 and >4000 that either didn't interfere with SAVE, or you used a modified SAVE, then you could save it together with the VDP utilities. However this is not necessary - you would just have two shorter files, and waste 1 or 2 sectors.

NOTE 2 High and Low Memory mix

If the file to be converted contains code below >4000 and above >A000 you need to convert the two parts separately, using the relocatable SAVE if necessary. If there is a low mem piece AND the utilities are also needed then for convenience you may want to save the entire low mem block together even though some space may be wasted. You may also do it in 2 separate pieces if you wish. In any case change the first 2 bytes of the file(s) to FFFF. Also remember that if you have 2 files, the second must have the last character increased by 1 (and again if there is code above >A000).

NOTE 3 Special File for Executable Instruction

If your first file created DID have an executable instruction at the beginning, AND you needed the special VDP utilities file, then change the latter to have a name AFTER the program files, and change its first two bytes back to 0000. Then you are set. If there wasn't an executable instruction and you do need the VDP file, then change bytes 7-10 of the VDP file to 0460xxxx where xxxx is where your program actually starts. You can do this because those 4 bytes were actually what CALL INIT loaded at >2000 to >2003 and aren't needed. One last case where you don't have the first executable instruction but don't have to make an extra file is where there was a BSS of at least 4 bytes at the start of the file (in other words, successive origins in the DIS/FIX 80 file). Then you can replace bytes 7-10 of your first program file with 0460 xxxx.

If none of these special cases obtain, then you will have to prepare this special file. It actually is rather easy. Find an unused area of memory, either in low mem between >2F00 and >4000, or high in the high mem, above >F000. Write the following source code:

```
      DEF  SFIRST,SLAST,SLOAD
      AORG >F000  OR WHEREVER YOU HAVE DECIDED IT GOES
SLOAD
SFIRST B    @>XXXX XXXX IS YOUR ACTUAL START ADDRESS
SLAST  END
```

Save this file, then assemble it, load it, then load SAVE, then press enter, type in SAVE, enter and follow the screen prompts. Use as your file name one with the last character one less than your previous first file. You will create a tiny 2 sector file which EA will find the start address in. Remember to use a sector editor to change the first 2 bytes from 0000 to FFFF.

NOTE 4 Multiple Files

If your program actually contained multiple files to load before the CALL LINK or the entry of program name, the instruction above still apply, but it may be a little harder to find the information you need. I'll be happy to help if I can, but you should be able to do it. Remember to load ALL of the files before running SAVE.

This article wound up a LOT longer than I intended. Unfortunately I have never been accused of being to brief. However I was really trying to cover all possibililites. I hope it works for you every time! Enjoy.

CLOSE ENOUGH part II

by Richard J. Bailey

Chick De Marti wrote an article entitled "CLOSE ENOUGH" that was published in the July, 1986 TopIcs. The routine in his article showed how you could check to see if some test number was within a certain percentage of a reference number. This is his routine.

```
100 CALL CLEAR
110 INPUT "A NUMBER ":TEST
120 PRINT : :
130 FOR I=20 TO 1 STEP -4
140 X=I/100 :: N=TEST-X
150 PRINT N
160 IF (N<TEST-.10) THEN GOSUB 250 ELSE GOSUB 270
170 NEXT I
180 PRINT
190 FOR I=1 TO 20 STEP 4
200 X=I/100 :: N=TEST+X
210 PRINT N
220 IF N>TEST+.10 THEN GOSUB 250 ELSE GOSUB 270
230 NEXT I
240 END
250 PRINT "SORRY"
260 RETURN
270 PRINT "CLOSE ENOUGH"
280 RETURN
```

I believe there is an easier way to implement this sort of check. Instead of using two separate checks to see if the number is greater than or less than some reference number, use the absolute value (ABS) of the difference and check to see if that is greater than some numeric value. All this can be done in one line (line 150 in my example) and has the general form: IF ABS(REFERENCE-TEST)>DIFFERENCE THEN GLOB1 ELSE GLOB2. GLOB1 and GLOB2 can be any legal command such as GOTO 250, GOSUB 899, PRINT "HELLO", etc., that makes sense in your program. GLOB2 may be optional depending on your needs.

In the original routine the check is to see if the TEST number is a certain numeric value of the reference. This is not true percent. For example, 100 plus/minus .1 and 1 plus/minus .1 don't have the same percentage difference. To get true percentage difference which the original routine didn't do, you need to define DIFFERENCE as: PERCENT*REFERENCE/100. 100 plus/minus (10*100/100) and 1 plus/minus (10*1/100) both check for a 10% difference.

Run and experiment with my routine to see what you can do with this idea.

```
100 ! + ++++++++++++++++ +
110 ! + CLOSE ENOUGH DEMO +
120 ! + ++++++++++++++++ +
130 CALL CHAR(126,"001010707C1010080 7C"):: DISPLAY
AT(7,4)ERASE ALL:"ANSWER MUST BE 20 ~ .1"
140 FOR I=1 TO 300 :: NEXT I :: X=RND*.4+19.8
:: DISPLAY AT(11,4)BEEP:USING "##.##":X
150 IF ABS(20-X)>.1 THEN DISPLAY AT(11,10):"DIF
FERENCE > .1" :: GOTO 140 ELSE DISPLAY AT (11,1
0):"CLOSE ENOUGH!" :: GOTO 140
```

Pre-Scan It

A Review by Howie Rosenberg

In one of the early issues of 99er Magazine (later Home Computer magazine) a letter to the editor written by C Ehninger of Futura Software appeared asking how HE COULD make the RUN statement embedded in a program accept literals ie RUN "DSK1."&A$. Mr. Ehninger had attempted to write an auto loader program to enable menu selection of Extended Basic programs on a disk which did not exist at that time but which abound today. In the very next issue, John Clulow had the solution. More important than the load program a whole new technique for Extended Basic programming on the TI-99/4 was born. The concept of using BASIC tokens and "poking" them into the code thus creating lines of program code was picked up first (by my recollection) by Barry Traver who has performed some miracles of his own using the technique. Peter Hoddie has taken the concept of operating on a program in merge format by means of adding tokens to new heights in his latest release Pre-Scan It.

How often have you typed RUN after booting an Extended Basic program and waited for an unreasonable time before the program started running? Most of the lost time is due to what is called the prescan during which time the code is scanned and space allocated for all variables. While provisions were made in Extended Basic version 110 for shortening this prescan time the process can be cumbersome for existing code particularly for long programs. Pre-Scan it allows you to take advantage of the prescan reduction without requiring you to modify the code yourself.

The usual method of reducing prescan is to add as many lines as necessary at the beginning of the program containing a list of all variables and calls used in the program. These lines need not be syntactically correct as the next step is to insert a jump around these lines with a GOTO statement. Finally the instructions !@P- and !@P+ are added to turn pre scan off and on as required. Once a line containing a variable is scanned it is not necessary to scan any other lines containing that variable. Generally it is only necessary to scan the few lines added at the beginning of the program. The result for long programs is a substantial reduction of that wait prior to program execution. With Pre-Scan It you simply save your program in MERGE format after making sure that as a minimum lines 1 to 9 are not used. If they are a simple RESequence will get you set. Next run Pre-Scan It and follow the prompts. It's as simple as that. The end result is a MERGE file which, when loaded in memory and saved in program format, is the original program with the prescan code added.

As this was intended as a review, and as reviews of computer software seem to all follow a prescribed format (I hate consistency) I will attempt to follow the format. I will not follow the traditional grade report card as, in my opinion, it is too subjectve and of little real value.

PERFORMANCE..I used Pre-Scan It on a variety of programs. It performed flawlessly. Originally I had version 1 which does have one bug in it. If your program contains Sub Programs which are on a single line, Pre-Scan It cannot insert the code to turn prescan off and on. This has been partially fixed in the latest version by giving the user the option of not operating on the Sub programs at all. For almost all cases this will not make any measurable difference in the prescan time. If you feel it to be necessary, you can change the original code by moving the SUBEND statement to a separate line which will then allow you to turn off prescan for the entire program. For the novice or non programmer, specifying no prescan removal for Subprograms from the menu is entirely sufficient. The program, written in Extended Basic runs rather slowly but as it is used only once on each program, the slow process is quite tolerable. Pre-Scan It does what it is intended for and does it well. Performance is EXCELLENT.

EASE OF USE..The program is well prompted. one quick reading of the documentation should suffice for even the novice user. The user must be aware of several options and the purpose of each, to enable him to make intelligent decisions when prompted for options. As an example, the progam optionally allows you to compress your program in several ways. This appears to have been incorporated in the program to allow adding the prescan code to very large programs as well as to speed up your program (not unlike SMASH in this respect). One of these features allows you to change up to five numbers with symbols such as @ which serves to reduce the memory space used. One must know whether the symbols have already been defined in the program in order to decide on whether this option should be exercised. The novice should have little problem in using the program. There are absolutely no problems associated with using this program.

DOCUMENTATION...The documentation originally supplied with Pre-Scan It was rather meager in several important areas and too verbose in others. Fortunately documentation is of little importance as the operation is so simple. I have seen an updated instruction manual which I understand will replace the original and which is excellent and extremely easy to read and understand.

VALUE..At $10 Pre-Scan It is unquestionably one of the best values ever for the TI-99/4A. A useful, professional, easy to use, utility.
My congratulations to Peter Hoddie for developing such a useful, well conceived, and competently coded utility.

Did you know that...?

by Chick De Marti

TRIVIA IN NUMBERS

1) The following number is totally unique. What's so special about it? --> 8,549,176,320 <--

2) Which weighs more, a pound of gold or a pound of lead?

3) What three different digits are represented by A,B,C in this addition problem?

```
  A C B
  A B C
  -----
  B A C
```

(Answers are on the next page.)

(*)(*)(*)(*)(*)(*)

This next bit of trivia comes from RANDY'S RUMOR RAG

Take a look at the LTA number on the bottom of your console. Ever wonder what that means?

The first two numbers stand for the week and the last two are the year your console was made. So, if yours says 1182, then it was made in the end of March in 1982.

(*)(*)(*)(*)(*)(*)

These next three items are from the FRONT RANGER Newsletter ODDS and ENDS by Bill Gaskill

(4) RUN can be used with CSi to redefine a character set in one program and RUN another program without "killing" the new character set or robbing memory from the second program being loaded?

(15) 4A/TALK can upload DV/80 files to CompuServe's TI FORUM in 7-bit ASCII? The results are screen readable but are NOT downloadable in readable format.

(18) A protected XB program on casette tape can be duplicated using two casette recorders set to maximum volume??

(*)(*)(*)(*)(*)(*)

If YOU have any "I Didn't Know That!" contributions or stories, send them to:

Chick De Marti
P.O.Box 3547
Gardena CA 47247-7247

Send 'em in any shape or form, I'll use it!

Anonymous. ON LITTLE THINGS

Most of us miss out on life's big prizes. The.. Pulitzer, Nobel, Oscars, Tonys and Emmys. But we're all eligible for life's small pleasures. A pat on the back. A kiss behind the ear. A four-pound Bass. A full moon. An empty parking space. A crackling fire. A great meal and a glorious sunset.

Don't fret about copping life's grand awards. Enjoy its tiny delights. There are plenty for all of us.

(*)(*)(*)(*)(*)(*)

Here's two items for your SO WHAT department!

You can load FORTH using the Miniman Cartridge. Use option 3 "Miniman" then Option 1 "LOAD AND RUN"

If you enter this short routine...

```
100 RANDOMIZE
110 FOR I=1 TO 10
120 N=INT(RND)
130 PRINT N;
140 NEXT I :: ! You'll get 10 zeros...
```

(*)(*)(*)(*)(*)(*)

This reminder from AMERICAL GROUP, Magnetic Media Distributors (see MicroTimes).

IS THE COMPUTER A FEMALE?

WHAT sex is a computer?

You may never have given the question much thought – imagining, perhaps, that a machine which makes thousands of decisions every minute hasn't got so much as a microsecond to spare for thoughts of the flesh. But you could be wrong.

For a start, why should a computer tell you every time it feels a sexy tingle in its peripherals? It's probably much too busy enjoying itself.

And – what is worse – it may not really fancy you. Can you honestly swear that you asked if it wanted to come and live with you?

Of course, anyone who has got beyond Section One of a computer manual will know how to make it say "I love you. Ready." Computers are always saying they are reading for something or other.

That's why Americans call them "user-friendly" and talk about "human/machine interfacing".

Go along to your local Dixons and you'll see them: rows of men (and boys) all busily interfacing with the computers.

You won't see many women there – and there's a reason. For computers are undoubtedly female.

Not that men admire computers for their looks. The petite, blonde Apple Mackintosh is the only one that comes near being an electronic Joanna Lumley.

So what do men see in them? For a start they have one of the virtues – if you can call it that – that men used to find in women. They are built for a life of drudgery.

Of course they won't iron shirts or darn socks, but that's only because those bright young inventors in California's Silicon Valley spend all their time in T-shirts and flip-flops. Clearly, though, a computer's place is in the home.

Once men got the idea that computers are a substitute for the uncomplaining housewife of old, they had to find ways of giving them commands.

So they came up with the computer language, BASIC, which stands for Brutish Arrogant Sexist Instruction Code.

BASIC has turned the computer into a man's plaything. Male chauvinist programmers can now have a quick peek or poke (not to mention a byte or a nibble) any time they feel the urge.

But what are real live women to make of this? Should they view computers as rivals for men's affections and go round sniping off their cables?

Do they bring out the worst in men, or are they just a harmless way of working off sexist tendencies?

That, as Sir Clive S would probably say, is the 64K question.

by ROGER GREEN News on Sunday
Caxton House, 13 Borough Road, SE1. (c)

—— from T*Times East Sussex, England ——

TI-WRITER INTO WORDSTAR

Edited from an article by Walt Maes in the Suncoast Beeper and later the K-TOWN 99ers

I have an IBM clone and a TI 99/4A sitting next to each other so I have interest in both machines. I have owned the TI for over two years and have many disks filled with text files. I don't want to retype all those files over again so that I can use them on the Wordstar. So, to transfer them, I found the first thing I needed was a null cable. The pin wiring is 2-2, 3-3 5-5, 6-20, 20-6, 7-7. I needed a male DB-25 for the TI and a female DB-25 for the clone. With the cable installed, I used QMODEM on the clone and Fast-Term on the TI. With Fast-Term loaded, I hit FCTN J to turn ON line feeds, and FCTN N to name the file I was sending. (On V1.16 of Fast-Term, hit FCTN-SHIFT T to make sure ASCII transfers are on, rather than TE II transfers - Ed.) Now go to QMODEM and press Home and change the baud rate. I used F for 1200 baud (Use CTRL 1 to change the baud rate on Fast-Term - Ed.) Then press Pg Dn and you are asked for the file name of the file you are going to transfer Then press FCTN , (FCTN comma) on Fast-Term to start the transfer. After the transfer, pull up Wordstar or PC-Write. To my surprise I found carriage returns on each line which I had to delete, as well as all TI-Writer commands. After doing that, it was ready to use.

(*)(*)(*)(*)(*)(*)(*)

DATABASE BUG .. Gary Cox has informed the users group (K/TOWN 99ers) that his Basic DATABASE program contains a small bug. The error is in line 1440 and 1750. Change "DSK1.DIR$ to "DSK1."DIR$.

(*)(*)(*)(*)(*)(*)(*)

(Answers to TRIVIA IN NUMBERS)

1 .. It is the only number that contains all the numerals in "alphabetical" order.

2 .. A pound of lead. (Lead is weighed by the standard measure of 7000 grains=one pound...gold is measured by troy weight, where one pound=5760 grains.

3 .. A = 4, B = 9, C = 5

(*)(*)(*)(*)(*)(*)(*)

Coffee's gone...
Break is over...
Back to the computer.
See you next month.

Chick

KIDS ##########

```
100 ! **********************
110 ! * BY STEVEN A SHOUSE
120 ! * RIVERSIDE CA 92506
130 ! *                    *
140 ! * TRANSLATED FROM A *
150 ! * TRS-80 BASIC PROG.*
160 ! **********************
170 CALL CLEAR
180 PRINT "       AROUND THE WORLD": :
190 PRINT "        IN EIGHTY DAYS": : : : : : : : :
200 FOR D=1 TO 750
210 NEXT D
220 C=0
230 PRINT "PHILEAS FOGG BET HE COULD GOAROUND THE WORLD IN 80 DAYS!"
240 PRINT :"WHY NOT JOIN HIM IN HIS HOT AIR BALLOON? HE'LL TELL YOU "
250 PRINT "WHAT HE SEES ON HIS TRAVELS AND YOU HELP HIM NAME THE"
260 PRINT "CONTINENT YOU'RE PASSING    OVER.  REMEMBER TO SPELL"
270 PRINT "THE CONTINENT CORRECTLY!": : : : :
280 GOSUB 1000
290 CALL CLEAR
300 READ N$,X$,Y$,Z$
310 IF N$="END" THEN 930
320 PRINT X$
330 PRINT
340 INPUT "Where are we? ":A$
350 PRINT :: CH=CH+1
360 IF A$=N$ THEN 540
370 PRINT "That's not it.Here's is an-":"other clue:":
380 PRINT
390 PRINT Y$:
400 PRINT
410 INPUT "Now,where are we? :":A$
420 PRINT :: CH=CH+1
430 IF A$=N$ THEN 540
440 PRINT :"That doesn't seem right...  Here is one more clue:":
450 PRINT
460 PRINT Z$:
470 PRINT
480 INPUT "Your last chance...Where arewe? ":A$
490 PRINT :: CH=CH+1
500 IF A$=N$ THEN 540
510 PRINT "LOOK! I see a sign that says""We're over ";N$;"!"
520 PRINT : :
530 GOTO 620
540 R=R+1
545 IF R>3 THEN R=1
550 ON R GOTO 560,580,600
560 PRINT :"That's it!!!": :
570 GOTO 610
580 PRINT :"ALL RIGHT! YOU GOT IT!"
590 GOTO 610
600 PRINT :"'WAY TO GO,CHAMP! YOUR RIGHT"
610 C=C+1
620 PRINT "Now we will move on to      another continent.": :
630 GOSUB 1000
640 GOTO 290
```

```
=============================
|                           |
| Slightly edited to make it |
| fit on the page.  P.S.  It |
| is also now in our library.|
|                           |
=============================
```

```
650 DATA EUROPE
660 DATA MY HOME COUNTRY OF ENGLAND  IS ON THIS CONTINENT.
670 DATA THE COUNTRIES ARE SMALL BUT THE ALPS ARE Soooo BIG
680 DATA I CAN SEE THE LARGEST CON-  TINENT OFF TO THE EAST
690 DATA AFRICA
700 DATA LOOK AT THOSE PEOPLE CANOE- ING DOWN THE NILE RIVER
710 DATA I SEE A VAST DESERT IN THE  NORTH AND A JUNGLE TO THE   SOUTH.
720 DATA I SEE A SIGN FOR EGYPT
730 DATA ASIA
740 DATA A SIGN POINTS THE WAY TO    MONGOLIA.
750 DATA WE'LL FLY OVER THE HIGHEST  MOUNTAIN IN THE WORLD.
760 DATA THIS CONTINENT GOES FOREVER!I'M SURE IT IS THE LARGEST.
770 DATA AUSTRALIA
780 DATA ABORIGINES WERE THE FIRST   TO SETTLE HERE.
790 DATA THIS HAS TO BE THE SMALLEST CONTINENT.
800 DATA WHAT WEIRD CREATURES! LOALA BEARS AND KANGAROOS.
810 DATA ANTARTICA
820 DATA BRRR! IT'S FREEZING DOWN    HERE!
830 DATA WE'RE ON THE BOTTOM OF THE  EARTH!
840 DATA LOOK! THERE'S THE SOUTH POLE
850 DATA SOUTH AMERICA
860 DATA THE ANDES MOUNTAINS LOOK    BEAUTUFUL DOWN THERE.
870 DATA THE AMAZON RIVER HAS TO BE  THE WIDEST IN THE WURLD.
880 DATA WE'RE SOUTH OF THE EQUATOR.
890 DATA NORTH AMERICA
900 DATA THERE'S GREENLAND TO THE    EAST.
910 DATA THIS CONTINENT STRETCHES ALLTHE WAY TO THE NORTH OPLE.
920 DATA LOOK HOW TALL AND WIDE THE  ROCKY MOUNTAINS ARE!
930 DATA END,END,END,END
940 CALL CLEAR
950 PRINT "PHILEAS MADE IT WITH ONE    SECOND TO SPARE!"
960 PRINT "THANKS FOR THE HELP": :
970 PRINT "BY THE WAY, YOU HAD";C;"OUT OF SEVEN CORRECT!"
980 PRINT "IN ONLY";CH;"TRIES!"
990 END
1000 PRINT "[Press <ENTER> to continue]"
1010 CALL KEY(0,K,S)
1020 IF S=0 THEN 1010
1030 RETURN
```

$ $ $ $

FORTH SCR #3

Wesley Richardson - Bluegrass 99 Computer Society

A comparison of different programming languages was started at the April meeting of the Society, and this article is a continuation of that theme.

Jon Keller wrote and demonstrated his SPIRAL program in c99 and in Extended BASIC. With the help of several people, his routine has been translated to four other languages. The TI-99/4A has the capability of running programs in c99, BASIC, Extended BASIC, UCSD Pascal, FORTH, LOGO, TMS9900 Assembly Language, PILOT 99, and possibly some others. The ease of writing the program, the ease of use, and the speed of execution varies widely between these languages. The run times for SPIRAL are given below:

EXTENDED BASIC	29m 11s
LOGO	21m 13s
USCD PASCAL	17m 20s
FORTH	1m 20s
c99	0m 58s
9900 ASSEMBLY	0m 16s

One of the criteria used in translating the program to the other languages, was to use the same algorithm in each program. Where possible, the names of the variables were also kept the same, to assist in correlation of the program listings.

To enter the c99 SPIRAL program, use the Editor/Assembler Editor, and save the program on disk as SPIRAL;C. Use option 5 of E/A to load C99C and answer 'Y' to the first two questions. Use SPIRAL;C as the input name and SPIRAL;S as the output. After compiling with no errors, load the E/A Assembler to assemble SPIRAL;S with no options and the output file SPIRAL;O. To run the program, load SPIRAL;O and load CSUP. The program calls dsk2.conio so this c99 file must be in disk drive 2. Give START as the program name to start.

```
        DEF  GO
        REF  VSBW
*********************************
*        SPIRAL PROGRAM         *
*   TI-99/4A ASSEMBLY LANGUAGE  *
* BLUEGRASS 99 COMPUTER SOCIETY *
*          APRIL 1986           *
*********************************
REGS    BSS  32
CN      EQU  9
ROW     EQU  7
COL     EQU  6
N       EQU  5
Z       EQU  8
D       EQU  10
X       EQU  4
Y       EQU  3
*********************************
GO      LWPI REGS
        LIMI >0000
        LI   CN,33

LIN160  LI   ROW,12
        LI   COL,16
        BL   @WRITEC
        LI   ROW,13
        LI   COL,17
        LI   N,2
        LI   Z,1
LOOPZ   LI   D,-1
        LI   X,1
LOOPX   LI   Y,1
LOOPY1  A    D,ROW
        BL   @WRITEC
        INC  Y
        C    Y,N
        JLE  LOOPY1
        LI   Y,1
LOOPY2  A    D,COL
        BL   @WRITEC
        INC  Y
        C    Y,N
        JLE  LOOPY2
        LI   D,1
        INC  X
        CI   X,2
        JLE  LOOPX
        INCT N
        INC  COL
        INC  ROW
        INC  Z
        LI   R2,12
        C    Z,R2
        JLT  LOOPZ
        INC  CN
        CI   CN,128
        JLT  LIN160
        LIMI 2
        BLWP @0
** WRITE CHARACTER ROUTINE **
WRITEC  LI   R0,32
        MPY  ROW,R0
        AI   R1,-32
        A    COL,R1
        AI   R1,-1
        MOV  R1,R0
        MOV  CN,R1
        SWPB R1
        BLWP @VSBW
        RT
        END
```

```
/* c99 SPIRAL PROGRAM SOURCE CODE */
/* BY JON KELLER */
/* BLUEGRASS 99 COMPUTER SOCIETY */
/* APRIL 1986 */

#define one 1
#define none -1
#include dsk2.conio

int row,col,cn;

main(){ /* MAIN PROGRAM */
  int ff,bp,cd,rd,x,y,n,c;
  cn=33;
  putchar(FF);
  while(cn!=126){
    n=2; row=14; col=16; x=1; bp=0;
    while(bp!=1){
      x=1;
      cd=none;
      rd=none;
      while(x<3){
        y=1;
        while(y<n){
          row=row+rd;
          if(row>24){row=24;bp=1;}
          hchar();
          y++;}
        y=1;
        while(y<n){
          col=col+cd;
          hchar();
          y++;}
        rd=one;
        cd=one;
        x++;}
      n=n+2;
      col++;
      row++;}
    cn++;}
  putchar(FF);
  puts("again ? Y or N \n");
  c=getchar();
  if ((c=='y')|(c=='Y'))main();

  } /* END OF MAIN PROGRAM */

hchar(){ /* HCHAR ROUTINE */
  locate(row,col);
  putchar(cn);
  }
return();
```

```
{* PASCAL SPIRAL PROGRAM *
{* BY WALTER LIGHTNER *}
{* BLUEGRASS 99 COMPUTER SOCIETY *}
{* APRIL 1986 *}
PROGRAM SPIRAL;
USES SUPPORT;
VAR ROW, COL, N, Z : INTEGER;
    RD, CD, X, Y : INTEGER;
    CN : INTEGER;
BEGIN
  PAGE(OUTPUT);
  CN:= 33;
    REPEAT
    ROW:= 13; COL:= 20;
    GOTOXY(19,12);
    WRITE(CHR(CN));
    N:= 2;
    FOR Z:= 1 TO 11 DO
      BEGIN
      RD:= -1;
      CD:= -1;
      FOR X:= 1 TO 2 DO
        BEGIN
        FOR Y:= 1 TO N DO
          BEGIN
          ROW:= ROW + RD;
          GOTOXY(COL,ROW);
          WRITE(CHR(CN))
          END;

        FOR Y:= 1 TO N DO
          BEGIN
          COL:= COL + CD;
          GOTOXY(COL,ROW);
          WRITE(CHR(CN))
          END;
        RD:= 1; CD:= 1
        END;
      N:= N +2;
      COL:= COL + 1;
      ROW:= ROW + 1
      END;
    CN:= CN + 1;
    UNTIL CN > 127;
END.
```

PROCEDURES

```
TO SPIRAL
; SPIRAL LOGO PROGRAM
; WESLEY R RICHARDSON
; BLUEGRASS 99 COMPUTER SOCIETY

; APRIL 1986
CS
MAKE "CN 33
REPEAT 63 [RESET LOOPZ MAKE "CN
 :CN + 1 ]
MAKE "CN 64
REPEAT 32 [RESET LOOPZ MAKE "CN
 :CN + 1 ]
END

TO RESET
MAKE "ROW 13
MAKE "COL 17
PT :CN 16 12
MAKE "N 2
MAKE "RO 1
MAKE "CO 1
END

TO LOOPZ
REPEAT 11 [MAKE "RO :RO - 2 MAK
E "CO :CO - 2 LOOPX MAKE "N :N
+ 2 MAKE "COL :COL + 1 MAKE "RO
W :ROW + 1 ]
END

TO LOOPX
REPEAT 2 [LOOPY1 LOOPY2 MAKE "R
O 1 MAKE "CO 1 ]
END

TO LOOPY1
REPEAT :N [MAKE "ROW :ROW + :RO
 CCHAR ]
END

TO LOOPY2
REPEAT :N [MAKE "COL :COL + :CO
 CCHAR ]
END

TO CCHAR
PT :CN :COL :ROW
END
** DONE **
```

```
SCR #31
 0 ( SPIRAL FORTH PROGRAM  -TEXT )
 1 ( WESLEY R RICHARDSON  APRIL 1986 )
 2 ( BLUEGRASS 99 COMPUTER SOCIETY )
 3 BASE->R DECIMAL : IT ;
 4 0 VARIABLE CN 0 VARIABLE ROW 0 VARIABLE COL
 5 0 VARIABLE  N 0 VARIABLE  RI 0 VARIABLE  CI
 6 : CCHAR COL @ ROW @ GOTOXY CN @ EMIT ;
 7 : LOOPY1 N @ 0 DO RI @ ROW +! CCHAR LOOP ;
 8 : LOOPY2 N @ 0 DO CI @ COL +! CCHAR LOOP ;
 9 : LOOPX 3 1 DO LOOPY1 LOOPY2 1 RI ! 1 CI ! LOOP ;
10 : LOOPZ 12 1 DO -1 RI ! -1 CI ! LOOPX 2 N +! 1
11   COL +! 1 ROW +! LOOP ;  ; RESET 12 ROW !
12   16 COL ! 15 11 GOTOXY CN @ EMIT 2 N ! ;
13 : SPIRAL CLS 33 CN ! BEGIN RESET LOOPZ 1 CN +!
14   CN @ 127 > UNTIL QUIT ;
15 SPIRAL   R->BASE
```

```
100 REM SPIRAL EXTENDED BASIC
110 REM JON KELLER
120 REM BLUEGRASS 99 COMPUTER SOCIETY
130 REM APRIL 1986
140 CALL CLEAR :: CN=33
150 ROW=13 :: COL=17
160 CALL HCHAR(12,16,CN)
170 N=2
180 FOR Z=1 TO 11
190 RO=-1 :: CO=-1
200 FOR X=1 TO 2
210 FOR Y=1 TO N
220 ROW=ROW+RO
230 CALL HCHAR(ROW,COL,CN)
240 NEXT Y
250 FOR Y=1 TO N
260 COL=COL+CO
270 CALL HCHAR(ROW,COL,CN)
280 NEXT Y
290 RO=1 :: CO=1
300 NEXT X
310 N=N+2 :: COL=COL+1 :: ROW=ROW+1
320 NEXT Z
330 CN=CN+1 :: IF CN<128 THEN 150
340 END
```

NEW PROGRAMS FROM THE LA 99 LIBRARY

Been busy adding **RLE** pictures this month 334 pictures on 37 disks. A picture of every picture on a disk will be aviable at the Library corner if you want to see what on a disk.

RLE-PICTURES RLE-PICTURES

PRICES ARE MARKED AND INCLUDES DISK, A TWO AND ONE/HALF PERCENT (2 1/2%) DISCOUNT WILL BE GIVEN FOR EVERY **ADDITIONAL DISK** ORDERED. :: 1=0%, 2=2 1/2%, 3=5%, 4=7 1/2%, 5=10%, 6=12 1/1%, 7=15%, 8=17 1/2%, 9=20%, 10=22 1/2%, 11=25%, 12=27 1/2%, 13=30%, 14=32 1/2%, 15=35%, 16=37 1/2%, 17=40%, 18=42 1/2%, 19=45%, 20=47 1/2%, 21 AND OVER = 50% MAX.

Include $1.00 for every TWO disks mailed. I will be putting your ordered on DSSD disks (flippies) That will = 4 RLE DISK PROGRAMS being mailed for $1.00. Number in ( ) are sectors used on disk.

2260 RLE #0 $5.00 MAX/RLE AND DOC. REQUIRED PROGRAM FOR VIEWING AND PRINTING ALL RLE PICTURES, REQUIRES 32K AND EDITOR ASSEMBLER MODULE, LOAD #3 DSK1.MAX/RLE -ENTER- PROGRAM NAME IS START-ENTER-. AFTER PROGRAM COME UP FOLLOW DIRECTION ON SCREEN.

2261 RLE #1 $5.00 (353) 1ST_EVA, 1WBUGS, 3_STOOGES, 3DVIEW, ABCNEWS, ALLEN-W, AMOXON, ANA, ANCHOR, ATFORD

2262 RLE #2 $5.00 (350) B-CROSBY, B-LUGUSI, BABY#1, BABY#2, BABY#2, BABY#7, BANANA, BARNEY

2263 RLE #3 $5.00 (350) BAMBI, BASKET, BATMAN, BATTLE, BEATLES, BEDROOM1, WOMAN, ZEN639

2264 RLE #4 $5.00 (356) BEDROOM2, BELKER, BEN, BILL_HERR, BIRDSFISH, BISSET, BLOOM, BLOOM/MILO, BLOOM/OLI, BLOOM/OPSI, O-JAY

2265 RLE #5 $5.00 (358) BLOOM/OPS2, BOGART-H, BREETA, BRINKLEY-C, BROWNOWL, BUNNY, CARRIB, VASQUEZ-R,

2266 RLE #6 $5.00 (356) BULLDOG, C-EASTWOOD, CARTUN, CASEY, CATS, CAVEGL, CBPIX1, SANTA, WEATHER

2267 RLE #7 $5.00 (347) CBPIX2, CBPIX3, CELTIC, CHAIR, CHALNGER, CHEKOV, CHESSMEN, CHIPMUNK

2268 RLE #8 $5.00 (350) CHRISTA, CHRISTI, CHRIS, CINDI, CITY, CLOWNS, COKE, COMPUTER, CONSOLE, CONSOL, CORVET, CURVE

2269 RLE #9 $5.00 (358) DALEK, DANGERFD-R, DASHERIFF!, DEAN-J, DEATHEAD, DENEUVE, DISNEY1, DISNEYLOGO, VAMPI

2270 RLE #10 $5.00 (358) DOG, DON/DUCK, DOUGLAS-K, DOUGLAS-M, DR-WHO, DRAGON, DRAGON#2, DRAGON1, DRAK, E-FLYNN, R-PRESLEY

2271 RLE #11 $5.00 (348) DRESSER, DROP, DUCK, DUNE, EAGLE, EANDME

2272 RLE #12 $5.00 (335) EASEL,EGRERT, EINSTEIN-A, EIREPARK, ELVIRA, ENTERPOOP, ENTERPTISE

2273 RLE #13 $5.00 (349) ENTERP, EVILFACE, FAIRLY1, FAIRLY2, FAIRLY3, FAIRLY4, FAIRLY5

2274 RLE #14 $5.00 (335) FBI1, FBI2, FBI3, FBI4, FBI5, FBI6, FBI7, FBI8, FERRI, FISH

2275 RLE #15 $5.00 (358) FEMWARRIOR, FIELDSWEET, FISH#1, FISH#2,FLAG, FLAMAP, FOXY, SATURNV, XMASTREE

2276 RLE #16 $5.00 (345) FLICKER, FLOWER, FROGGER, GADWALL, GATYPE, GBUST, GERSHEP,GETSOLRMAX

2277 RLE #17 $5.00 (354) (X-RATED DISK) FLUFFY, GIRLB, GIRLS1, GIRLS2, GILRS3, GIRLS4, GIRLS5, GIRLS6, MURPHY, NUDE3, NUDW4, NUDES, NUDE

2278 RLE #18 $5.00 (356) GLOVAL, GODZILLA, GOOFY, GREEKRUINS, HANDSDRAW, HENRY-K, HILLSTRO1, HILLSTRO2, HORSEBK, HOUSE, HOWDY, SCRENERY

2279 RLE #19 $5.00 (336) HUNTER, I-SPY, INDIAN, J-MANSFLD, JENNA, JOYLEAF, KADAFY, KALIOP, KARATE

2280 RLE #20 $5.00 (356) KENT-C, KHAN!, KING&ALICE, KOJO, L&H, L-NEMDY, LAUNCH, LEDZEP, WALL, WARGIRL2, WEEKND

2281 RLE #21 $5.00 (351) LADIES, LEM, LINDA, LINDA1, LOGO, LYBOIDS, MAC, MADONNA, MAMMOTH, MARLEN, W-DISNEY

2282 RLE #22 $5.00 (340) MATHART, MCAULIF, MCCOY, MERCURY, MERLIN, MEW, MICKEY, MICKEY2, MIDDLE

2283 RLE #23 $5.00 (356) MILO, MIMI-4, MINMAYDOLL, MMU, MOONWALK, MORGAN, MOUSE, MOUSEL, WASHINGTN-G

2284 RLE #24 $5.00 (358) MUSIC, NASAMAP, NATALIE, NEEDLE, NEUMAN-AE, NEWMAN, NICHOL, NINJATURTL, OCEAN, SPY

2285 RLE #25 $5.00 (355) OLIVER, ONI-MASK, OPUS, OPUS2, ORGUSS, ORGINAL, OUI/COVER, PAGODA, PAM, PELICIAN, PENGUIN, PLAYBOY, WEATHMAP

2286 RLE #26 $5.00 (339) POPCHASER, PORCUOINE, PYRAMID, QUAIL, R-MAN, R-REAGAN, R/HORROR, RADIOTEL

2287 RLE #27 $5.00 (357) RAILING, RALPH, RICK&LINDA, RICK1, RICK2, ROBORANDI, ROBOTECH04

2288 RLE #28 $5.00 (358) ROBOTCHOS, ROCKETS, ROCOT, RODNEY, ROY, RUINS, SAILBOARD, SALLY, SAM-EAGLE, SARAH-F

2289 RLE #29 $5.00 (337) SARGEANT, SATURN, SCOTT, SCROOG, SDF1, SEER, SELLEVK, SHEKKS, UWATER, UZI

2290 RLE #30 $5.00 (346) SHEPARD, SHIELDS-B, SHOW, SHREDDER, SNOOPY1, SNOPY2, SNOOWL

2291 RLE #31 $5.00 (331) SHUTTLE1, SHUTTLE2, SHUTTLE3, SHUTTLE4, SKYLAB, SPACE2, SPACEARM, SPACETEL, SPACEWALK

2292 RLE #32 $5.00 (294) SPCSTATION, SPCWLK, STAR1, STAR2, STAR3, STAR4, STARTREK05, STARWAR02, UHURA, WOOKIE

2293 RLE #33 $5.00 (345) SPRING, STALONE, STAR10X, STEVENICKS, STREET, STRIPE, WENDY-O, WALES

2294 RLE #34 $5.00 (212) (X RATED DISK) SCCUBS, SUMMIT, SUSAN, VENUPIC, WINDOW

2295 RLE #35 $5.00 (357) SUN, SWASH, SWDFIGHT, TAYLOR-E, TEMPLE-S, THEKING, TIGER1, TIGER2, TIGER3, TINA

2296 RLE #36 $5.00 (349) TKZ, TMNT, TMNT2, TWOONARM, UGLYALIEN, UNCLEBUNNY,USEMAP

2297 RLE #37 $5.00 (270) VAMPIRE1, VAMPIRE2, VF1A-SHIP, VIPER, WOODLAND

LIBRARIAN FRED MOORE 7730 EMERSON AVE. LOS ANGELES,CA 90045

MARKETPLACE

(the marketplace is a fund raiser for the club, that is, the "profit" goes to maintain the quality of this Newsletter. In general the price listed splits the difference between cost and retail. Please help your Club.)

MILLERS GRAPHICS	
DISKASSEMBLER	18.50
ORPHAN CHRONICLES (priceless)	9.95
ADVANCED DIAGNOSTICS	18.50
EXPLORER	22.50
NIGHT MISSION	18.50
GRAM KRACKER (80K EXPANDED)	185.00
GK UTILITY I	10.00
SMART PROGRAMING FOR SPRITES	6.25
NEW RELEASES	
PRE-SCAN IT! (J.PETER HODDIE)	10.00
GRAM PACKER "	10.00
FONT WRITER "	19.00
PRINTER'S APPRENTICE (M.McCANN)	19.00
MYARC	
RS232	82.00
D/D DISK CONTROLLER	155.00
128K RAM DISK/SPOOLER	175.00
512K RAM DISK/SPOOLER	280.00
EXTENDED BASIC II LEVEL IV	80.00
128K RAM DISK W/XBASIC II	235.00
512K RAM DISK W/XBASIC II	340.00
INSCEBOT	
TI-ARTIST	17.00
DISPLAY MASTER	12.00
ARTIST EXTRAS	6.00
MEGATRONICS	
EXTENDED BASIC II PLUS	72.50
INTERN (BOOK ON GPL)	16.50
128K GRAM CARD	227.50
HARDWARE & SUPPLIES	
TEAC 55BV DSDD DRIVES	110.00
DISKETTES DSDD	1.00
64K EPSON INT. PRINT BUFFER	45.00
COLOR RIBBONS (EPSON)	4.00
BACK ISSUES	
SUPER 99 MONTHLY	1.25
MICROPENDIUM	1.25
SMART PROGRAMMER JUNE 1986	1.50
BEST OF NEWSLETTERS W/DISK	5.00
FORTH NOTES VOL 1-5 (2.50 EA)	10.00
BEGINNER'S FORTH NOTEBOOK	2.50
ASSEMBLY NOTES VOL 1	2.50
TECHNICAL AND BUSINESS BOOKS	5.00
SAMS BOOKS (VARIOUS)	5.00
SAMS BOOKS WITH CASSETTES	7.50

CLUB MEETINGS

Los Angeles 99er Computer Group: Fourth Wednesday of each month, 7:15 PM, at Torrance Library, 3031 Torrance Blvd., Torrance.

Pomona Valley 99ers Computer Group: Second Tuesday of each month, 7:00 PM, at Cable Airport Cafe, 13th & Benson, Upland. Call Joy Warner, 982-9971, nights.

San Fernando Valley 99er Computer Group: Second Tuesday of each month, 7:30 PM, Doctors' Conference Room, Sherman Oaks Community Hospital, 4929 Van Nuys Blvd., Sherman Oaks.

San Gabriel Valley 99/4 Users Group: First Wednesday of each month at West Covina Public Library, 1601 W. Covina Parkway, West Covina.

Users Group of Orange County: Third Thursday of each month, 7:30 PM at Westchester Community Service Center (1 block east of Beach Blvd.), Jackson and Westminster, Westminster, CA.

Executive Board meeting of the Los Angeles 99ers is held at 7:15 PM, the first Tuesday of each month at Merit Savings Bank, 18501 S. Western Ave., Torrance. ** ALL MEMBERS ARE INVITED **

The first line on the address label shows the last issue you will receive for members or the last issue received by us for exchanges.

Board Meetings for the LA 99ers is held on the 1st Tuesday of each month at MERIT SAVINGS at 18501 Western Ave. Gardena The meeting starts at 7:15. *** EVERYONE IS INVITED ***

XXXXXXXXXXXXXXXXXXXXXXXXX
FIRST CLASS
XXXXXXXXXXXXXXXXXXXXXXXXX

Exchng * Oct 86
Dallas TI Home Computer Group
1221 Mosswood Place
Irving, TX 75061

oooooooooooooooooooooooooooooooooo

99'FEST-WEST'86

LA 99ers Computer Group
P.O. Box 3547
Gardena, CA 90247-7247

oooooooooooooooooooooooooooooooooo